TEXAS A&I UNIVERSITY

Kingsville, Texas

TECHNICAL REPORT #2

COMPUTER PROGRAMS

AND DOCUMENTATION

DEPARTMENT OF

MATHEMATICS

TEXAS A&I UNIVERSITY

RESEARCH GRANT #NGR 44-073-003

TECHNICAL REPORT #2

COMPUTER PROGRAMS AND DOCUMENTATION

F. M. Speed

S. L. Broadwater

June, 1971

DEPARTMENT OF MATHEMATICS

TABLE OF CONTENTS

1. INTRODUCTION

This report contains a description of the various statistical tests that were used to check out random number generators. The tests contained in this report are by no means all the possible tests that can be run. A total of 12 different tests were considered. And from these, 6 were choosen to be used. Among those not included in this report are such tests as the poker test, the coupon test, the spectral test, and so on. The 6 tests that were choosen were done so because of the properties that they appeared to exhibit. Also, these are the most classical tests that are run. One test, which was not included, is the spectral test. The reason why it was not included was because of the need of a very large computer. If such a computer would have been available, we would have included this test because it is a very powerful test.

The tests included in this report are the frequency test, the max t test, the run test, the lag product test, the gap test, and the matrix test. This report is divided into three major sections. The first section concerns those tests of goodness of fit; and under this we have the frequency and the max t test. The next section consists of those tests of independence; and this includes the run, the lag product, the gap, and the matrix test. The final section gives documentation on the use of these various tests as well as a listing of the programs.

The discussion in parts 2 and 3 makes the following assumptions. We have a sequence U_1, U_2, U_3, and so on that come from a pseudorandom number generator that is supposed to be generating random numbers from a uniform distribution and the numbers are supposed to be independently distributed. For the remainder of this report the terminology "random numbers" will be used to mean pseudorandom numbers.

2. GOODNESS OF FIT TESTS

A) Frequency Tests

The frequency test is one of the most popular tests used to check the uniformity of sequence of numbers. It consists in dividing the unit interval (0,1) into k equal subintervals. Then a sequence of N pseudorandom numbers are generated. The number that fall in each of the subintervals is calculated and a chi-square test is applied. The chi-square test consisting of the observed number in each subinterval minus the expected number. The expected number in each interval is simply N/K. The distribution of the sum of the observed minus the expected squared divided by expected is approximately a chi-square with k - 1 degrees of freedom. It should be noted however that the expected number in each subinterval should be greater than five.

B) The Max T Test

In order to use the max t test, the following sequence is obtained:

$$S_j = \max(U_{j1}, U_{j2}, \ldots, U_{jt})$$

It will be shown that S_j has distribution function $F(s) = s^t$. Hence the Kolmogorov-Smirnov test can be used with $F(s) = s^t$. To see that $F(S) = s^t$, let us consider

$$F(S) = P\{S_j \le s\} = P\{\max(U_{j1}, \ldots, U_{jt}) \le s\}$$

since the maximum is less than or equal to S. Hence

$$F(S) = P\{U_{j1} \le s, U_{j2} \le s, \ldots, U_{jt} \le s\}$$

$$= \prod_{i=1}^{t} P\{U_{ji} \le s\} \quad \text{since all } U_{ji} \text{ are independent.}$$

But $P\{U_{ji} \le s\} = s$, since $U_{ji} \sim U(0,1)$. Thus $F(S) = s^t$.

3. TESTS FOR INDEPENDENCE

A) The Run Test

A sequence of numbers may be tested for runs up or may be tested for runs down by examining the length of monotone subsequences of the original sequence. That is, we investigate segements which are either increasing or decreasing. As an example of a run test, let us consider the following sequence 5,4,1,2,9,6,3,4,5,2,1,5,4 in this sequence, we have 4 runs of length 1, 1 of length 2, and 2 of length 3. Note that contrary to the way most run test have been conducted, the chi-square test should not be applied to this data since the adjacent runs are not independent. Instead we shall use this data to construct a chi-square test that can be applied.

Let A be a 6x1 vector such that a_i = number of runs of length i, i = 1,2,...,5 and a_6 = number of runs of length six or more. Let B be a 6x1 vector such that $E(a_i)$ =

b_i and let C be 6x6 matrix such that V(A) = C, i.e. C is the covariance matrix of A. It has been shown in the "Annals of Mathematical Statistics" Vol. 15, p. 163-165, that A becomes normally distributed as the length of the original sequence tends to infinity. Thus $(A - B)^T C^{-1} A - B$ is approximately chi-square with 6 degrees of freedom. Expression for B and C are given in chapter 4.

B) Gap Test

This test is used to measure the lengths of n gaps. In this test, random numbers are generated until n gaps occur. In this test we used a gap size of .1. (Note a gap size of .5 is equivalent to test of runs above or below the mean). Let A be a tx1 vector such that

a_i = number of gaps of length i, i = 1,...,t-1

$\bar{a}_t$ = number of gaps of length t or greater.

Generate random numbers until $A_i = NG$, then B is a tx1 vector of expected values; i.e. $b_j = NG \cdot P_j$ where $P_j = q(1-q)^{j-1}$ $j = 1,\ldots, t-1$; $P_t = (1-q)^t$ and where q = .1. Hence

$$\chi^2 = \sum_{j=1}^{t} (a_j - b_j)^2/b_j,$$

which is chi-square with t - 1 degrees of freedom. Note we must choose NG and t so that $b_j \geq 5$ for $j = 1,\ldots, t$. The derivation of P_j is as follows:

The probability of a gap of length 1 means that a number must be followed by itself. The probability that it occurs is just q. The probability of a gap of length 2 means we must have a number followed by a different number and then followed by itself. Hence the probability that this happens is $(1 - q)q$ etc.

C) The Lagged Product Test

This test is used to determine if there is a correlation between U_i and U_{i+k}, where $k = 1, 2, \ldots$. In our test, $k = 1, \ldots, 10$. The following statistic was computed:

$$C_k = \frac{1}{N - k} \sum_{i=1}^{N-k} U_i \cdot U_{i+k}$$

If N is large and if there is no correlation between U_i and U_{i+k}, then C_k is approximately normally distributed with $E(C_k) = .25$ and $V\left[C_k\right] = (13N - 9k)/144(N - k)^2$. This can be seen from the following:

$$E\left[C_k\right] = \frac{1}{N - k} \sum_{i=1}^{N-k} E(U_i \cdot U_{i+k}).$$

But U_i and U_{i+k} are assumed uncorrelated with means equal to .5 . Hence $E(C_k) = \frac{1}{N - k} \sum_{i=1}^{N-k} (.5)(.5) = .25$

$$V(C_k) = \frac{1}{(N - k)^2} V\left[\sum_{i=1}^{N-k} U_i U_{i+k}\right]$$

$$= \frac{1}{(N - k)^2} \sum_{i=1}^{N-k} V(U_i U_{i+k}) + 2 \sum_{i}^{N-k} \sum_{j} \text{Cov} (U_i U_{i+k}, U_j U_{j+k}$$

But $V\left[U_i U_{i+k}\right] = E\left[U_i U_{i+k}\right]^2 - (E(U_i U_{i+k}))^2$

$$= E\left[U_i^2\right] E\left[U_{i+k}\right]^2 - (1/4)^2$$

$$= 4/12 \cdot 4/12 - 1/16$$

$$= 16/144 - 9/144 = 7/144$$

$$\mathrm{Cov}(U_iU_{i+k},U_jU_{j+k}) = E\left[U_iU_{i+k}\ U_jU_{j+k}\right] - E(U_iU_{i+k})E(U_jU_{j+k})$$

If $j \neq i + k$, then $\mathrm{Cov}(U_iU_{i+k},U_jU_{j+k}) = 0$. If $j = i + k$, then

$$\mathrm{Cov}(U_iU_{i+k},U_jU_{j+k}) = E(U_{i+k})^2 1/4 - 1/16$$

$$= 4/12 \cdot 1/4 - 1/16 = 1/12 - 1/16$$

$$= 3/144$$

They are $N - 2k$ times that $i + k = j$. Hence

$$V\left[C_k\right] = \frac{1}{(N-k)^2}\left[(N - k)\ 7/144 + (N - 2k)\ 6/144\right]$$

$$= \frac{1}{(N-k)^2 \cdot 144}\left[7N - 7l + 6N - 12k\right]$$

$$= \frac{1}{(N-k)^2 \cdot 144}\left[13N - 19k\right]$$

D) Matrix Test

In order to investigate the degree of randomness between successive numbers in a sequence the matrix test was employed. This test was proposed by M. L. Tuncosa and suggests one construct a k by k matrix whose elements x_{ij} represent the number of times a number in the i^{th} interval is followed by a number in the j^{th} interval. A sequence of M consecutive sets of N random numbers is generated, and equal values are expected for all the matrix elements. The chi-square statistic

$$\chi^2 = \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{(x_{ij} - N/k^2)^2}{N/k^2}$$

is computed and compared with expected chi-square distribution with $k^2 - 1$ degrees of freedom. A 90 per cent confidence interval was established as in the frequency test and is 948.1 and 1097.9. All generators with chi-square values in this range were considered acceptable.

4. SUBROUTINES

This section contains the subroutines used to carry out the tests. These were written in Fortran IV and run on the IBM 360/44.

4.1 Goodness of Fit Tests

a) Frequency Test

b) Max T Test

4.2 Tests for Independence

a) Run Test

b) Gap Test

c) Lagged Product Test

d) Matrix Test

SUBROUTINE FREQ

SOURCE:

Naylor, T. H., Balintfy, J. L., Burdick, D. S., and Chu Kong. Computer Simulation Techniques. New York: John Wiley and Son, Inc., 1966.

PURPOSE:

To check the uniformity of the distribution of the N random numbers.

CALLING SEQUENCE:

Random numbers between 0 and 1 are already generated and divided into J groups before FREQ is called.

CALL FREQ (COUNT, J, N)

where:

N is the number of random numbers

J is the number of groups that the random numbers have been divided into.

COUNT is an array that contains the number of random numbers in each group.

METHOD:

The statistic χ^2 is computed by:

$$\chi^2 = \sum_{I=1}^{J} (\text{COUNT}(I) - E)^2/E$$

where E is the expected number of random numbers in each group. χ^2 has approximately a chi-square distribution with J - 1 degrees of freedom for a sequence of "truly" random numbers.

COMMENTS:

This subroutine calculates the upper and lower limits. Z and w are the chi-square values at 90% confidence interval with J - 1 degeees of freedom. The percent of the confidence interval may be changed, by changing z and w in the subroutine.

```
      SUBROUTINE FREQ (COUNT,J,N)
      DIMENSION COUNT(1)
C  IN1 AND IN2 ARE LOGICAL DEVICE NUMBERS.  TEXAS A+I
C USES 1 TO READ AND 3 TO WRITE FOR THE IBM 360/44 COMPUTER.
      IN1=1
      IN2=3
      E=FLOAT(N)/FLOAT(J)
      CS=0.
      DO 4 I=1,J
      CHI=((COUNT(I)-E)**2)/E
    4 CS=CS+CHI
      WRITE(IN2,11) CS
   11 FORMAT (1H ,F15.6)
      W=-1.64
      Z=1.64
      K=J-1
      A=W*SQRT(2.*K)+K
      B=Z*SQRT(2.*K)+K
      WRITE(IN2,10) K,A,B
   10 FORMAT (1H ,10X,'90( CONFIDENCE INTERVAL WITH K DEGREES OF FREEDOM
     1',/,10X,'K = ',I8,'A = ',F15.3,'B = ',F15.3)
      IF (CS .GE. A .AND. CS .LE. B) GO TO 22
      WRITE(IN2,21)
      GO TO 88
   22 WRITE(IN2,26)
   21 FORMAT (1H ,'REJECT FREQUENCY DISTRIBUTION')
   26 FORMAT (1H ,'ACCEPT FREQUENCY DISTRIBUTION')
   88 RETURN
      END
```

SUBROUTINE KOLSMR

SOURCE:

Knuth, Donald E., The Art of Computer Programming. Addison-Wesley Publishing Company, Inc. 1969.

PURPOSE:

To determine if the random numbers come from a specified distribution.

CALLING SEQUENCE:

CALL KLOSMR (V,N,F,KN,D)

where:

V is an array containing the N random variables.

N is the number of random variables.

F is a function defined as $F(x) = x^t$, where t is the number of random numbers used to compute each V_i.

$$KN = \begin{cases} 0 & \text{if } u = 1,\ s = 1 \\ 1 & \text{if } u - 0,\ s \text{ is calculated} \\ 2 & \text{if } u \text{ and } s \text{ are calculated} \end{cases}$$

$D = MAX|F(x) - S_n(x)|$ see method below

METHOD:

N observations of the random quantity, X are obtained. The observations are rearranged so that they are sorted into ascending order, i.e., so that $X_1 \leq X_2 \leq \ldots \leq X_n$.

The following statistic is computed:

$$D = MAX|F(x) - S_n(x)|$$

where $F(x)$ - probability that $X \leq (x)$.

$$S_n(x) = j/n, \quad X_j \leq x \leq X_{j+1}$$

This value, D, is compared to a critical value in a table to determine if the data came from a specified distribution.

```
      SUBROUTINE KOLSMR(X,N,F,KN,D)
      DIMENSION X(N),G(2000),JJ1(200)
C   IN1 AND IN2 ARE LOGICAL DEVICE NUMBERS.  TEXAS A+I
C USES 1 TO READ AND 3 TO WRITE FOR THE IBM 360/44 COMPUTER.
      IN1=1
      IN2=3
      H=N
      IF(KN-1) 2,3,5
    2 U=0.
      V=1.0
      GO TO 8
    3 U=0.
      SQ=0.
      DO 4  I=1,N
    4 SQ=SQ+X(I)**2
      V=SQRT(SQ/N)
      GO TO 8
    5 S=0.
      SQ=0.
      DO 7  I=1,N
    7 S=S+X(I)
      U=S/FLOAT(N)
       DO 6  I=1,N
    6  SQ=(X(I)-U)*(X(I)-U)+SQ
       V=SQRT(SQ/FLOAT(N-1))
    8 WRITE(IN2,899) U,V
    9 CALL ORSEG(X,G,N,JJ1)
      DO 661  I=1,N
  661 X(I)=G(I)
      E=.1E-07
      D=0.
      DO 48  I=1,N
      B=I
      X1K1=(X(I)-U)/V
      FOB1=B/H
      FOB=(B-1.)/H
      FXK  =F(X1K1*(1.0-E))
      FXK1=F(X1K1*(1.0+E))
      Z=FOB-FXK
      Z1=FOB1-FXK1
      Y=ABS(Z)
      Y1=ABS(Z1)
      IF(Y1.GT.Y) GO TO 22
       IF(Y.GT.D) D=Y
      ZM=Z
      GO TO 48
   22  IF(Y1.GT.D) D=Y1
      ZM=Z1
   48 CONTINUE
      WRITE(IN2,901) D
  899 FORMAT(6HOMEAN=,F16.6,4X,5HS.D.=,F16.6)
  901 FORMAT(21HOMAXIMUM DEVIATION IS,F8.5)
      RETURN
      END
```

```
FUNCTION F(X)
COMMON IT
F=X**IT
RETURN
END
```

```
      SUBROUTINE ORSEG( X,Y,N,J)
      DIMENSION X(1), Y(1)
      DIMENSION J(1)
      IF(N.LE.20) GOTO 23
      IF(N.LE.130) GOTO 25
      NS=.014*FLOAT(N)+7.68
      INT=1
      NS1=NS-1
      DO 10 I=1,NS
      IP=I-1
   10 CALL ORDER2(X,N,NS,IP)
      M=0
      J(1)=1
      DO 11 I=2,NS
   11 J(I)=(I-1)*(N/NS)+1
      K1=2
   18 DO 13 I=K1,NS
      M1=J(I)
      M2=J(I-1)
      IF(X(M1).LT.X(M2)) GOTO 12
   13 CONTINUE
      GOTO 14
   12 DO 15 K=1,NS
      M3=J(K)
      IF(X(M1).LT.X(M3)) GOTO 16
   15 CONTINUE
   16 CONTINUE
      I1=I-1
      DO 17 KJ=K,I1
      J1=K+I1-KJ
      M4=J(J1)
      J(J1)=J(J1+1)
   17 J(J1+1)=M4
      K1=I+1
      IF(K1.GT.NS) GOTO 14
      GOTO 18
   14 M=M+1
      CHK2=NS
      NP=J(INT)
      Y(M)=X(NP)
      IF(M.EQ.N) RETURN
      J(INT)=NP+1
      IF(J(INT).LE.(NS-1)*(N/NS)+1) GOTO 21
      IF(J(INT).LE.N) GOTO 20
   22 INT=INT+1
      GOTO 20
   21 IF(MOD((J(INT)-1),N/NS).EQ.0) INT=INT+1
   20 J1=J(INT)
      I3= INT+1
      IF(I3.GT.NS) GOTO 14
      DO 19 KQ=I3,NS
      CHK3=NS
      J2=J(KQ)
      IF(X(J1).LT.X(J2)) GOTO 14
      J(KQ)=J(KQ-1)
      J(KQ-1)=J2
   19 CONTINUE
      GOTO 14
   23 CALL ORDER2(X,N,1,0)
```

```
      DO 24 I=1,N
   24 Y(I)=X(I)
      RETURN
   25 CALL ORSEG3(X,Y,N)
      RETURN
      END
```

```
      SUBROUTINE ORSEG3(X,Y,N)
      DIMENSION X(1), Y(1)
      N3=N/3
      DO 10 K=1,3
      IP=K-1
   10 CALL ORDER2(X,N,3,IP)
      M=0
      I=1
      J=N3+1
      K=2*(N/3)+1
      IF(X(I).LT.X(J)) GOTO 14
      NS=I
      I=J
      J=NS
   14 IF(X(J).LT.X(K)) GOTO 13
      IF(X(K).GT.X(I)) GOTO 15
      NS=I
       I=K
      K=NS
   15 NS=J
      J=K
      K=NS
   13 M=M+1
      Y(M)=X(I)
      IF(M.EQ.N) RETURN
      IF(I.EQ.N) GOTO 16
      I=I+1
      IF(X(I).LT.X(I-1)) GOTO 16
      IF(X(I).LT.X(J)) GOTO 13
      NS=I
      I=J
      J=NS
      IF(X(J).LT.X(K)) GOTO 13
      GOTO 15
   16 M=M+1
      Y(M)=X(J)
      IF(M.EQ.N) RETURN
      J=J+1
      IF(X(J).LT.X(J-1)) GOTO 18
      IF(X(J).LT.X(K)) GOTO 16
   17 M=M+1
      Y(M)=X(K)
      IF(M.EQ.N) RETURN
      K=K+1
      IF(X(K).LT.X(K-1)) GOTO 20
      IF(X(K).LT.X(J)) GOTO 17
      GOTO 16
   20 K=J
   18 M1=M+1
      DO 19 MS=M1,N
      Y(MS)=X(K)
   19 K=K+1
      RETURN
      END
```

```
      SUBROUTINE ORDER2(X,N,L1,L2)
      DIMENSION X(1)
      N2=L2*(N/L1)+1
      NN=(L2+1)*(N/L1)
      IF(L2.EQ.L1-1) NN=N
      K1=N2+1
    4 DO 99 I=K1,NN
      IF(X(I).LT.X(I-1)) GOTO 76
   99 CONTINUE
      RETURN
   76 DO 82 K=N2,NN
      IF(X(I).LT.X(K)) GO TO 84
   82 CONTINUE
   84 Z=X(I)
      I1=I-1
      DO 86KJ=K,I1
      J=K+I1-KJ
   86 X(J+1)=X(J)
      X(K)=Z
      K1=I+1
      IF(K1.GT.NN) RETURN
      GOTO 4
      END
```

MAX "T" TEST

N random variables are generated where each random variable, V_i, is defined as:

$$V_i = \text{MAX}\ (R_{1i}, R_{2i}, \ldots, R_{ti}).$$

Each R_{ji} is a random number, where $j = 0, 1, 2, \ldots, t$, and t is the number of random numbers.

The Kolmogorov - Smirnov test is applied to the sequence $V_0, V_1, \ldots, V_{n-1}$, with the distribution function $F(x) = x^t$, $(0 \leq x \leq 1)$.

SUBROUTINE RUN

SOURCE:

Knuth, Donald E., The Art of Computer Programming. Addison-Wesley Publishing Company, Inc. 1969.

PURPOSE:

To determine if the length of runs come from "true" random numbers.

CALLING SEQUENCE:

N random numbers between 0 and 1 have been generated before RUN is called.

CALL RUN(N,R,A)

where:

- N is the number of random numbers.
- R is an array containing the random numbers.
- A is an array containing the coefficients used to compute the statistic V.

METHOD:

The length of runs are determined. The length of a run is the number of consecutive increasing numbers inclusively. Any run longer than six (6) is counted as a run of 6.

The statistic V is then computed by:

$$V = \frac{1}{N} \sum_{I=1}^{6} \sum_{J=1}^{6} (\mathrm{COUNT}(I) - N*B(I))*(\mathrm{COUNT}(J) - N*B(J)*A(I,J))$$

where the coefficients A(I,J) and B(J) are:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} = \begin{bmatrix} 4529.4 & 9044.9 & 13568 & 18091 & 22615 & 27892 \\ 9044.9 & 18097 & 27139 & 36187 & 45234 & 55789 \\ 13568 & 27139 & 40721 & 54281 & 67852 & 83685 \\ 18091 & 36187 & 54281 & 72414 & 90470 & 111580 \\ 22615 & 45234 & 67852 & 90470 & 113262 & 139476 \\ 27892 & 55789 & 83685 & 111580 & 139476 & 172860 \end{bmatrix}$$

$$(b_1\ b_2\ b_3\ b_4\ b_5\ b_6) = \left(\frac{1}{6}\ \frac{5}{24}\ \frac{11}{120}\ \frac{19}{720}\ \frac{29}{5040}\ \frac{1}{840}\right)$$

V should have the chi-square distribution with six degrees of freedom.

COMMENT:

The upper and lower limits for a 90% confidence interval have been put into the subroutine. V is then checked to see if it falls between these limits. The percent of confidence interval may be changed by changing the limits in the subroutine.

```
      SUBROUTINE RUN (N,T,A)
      INTEGER COUNT(20)
      DIMENSION T(1),B(6),A(6,6)
C   IN1 AND IN2 ARE LOGICAL DEVICE NUMBERS.  TEXAS A+I
C USES 1 TO READ AND 3 TO WRITE FOR THE IBM 360/44 COMPUTER.
      IN1=1
      IN2=3
      IA=6
      DO 2 I=1,IA
    2 COUNT(I)=0
      T(N)=0.0
      R=1
      DO 18 J=1,N
      IF(T(J) .GT. T(J+1)) GO TO 22
      R=R+1
   18 CONTINUE
   22 IF (R .GE. IA) GO TO 24
   23 COUNT(R)=COUNT(R)+1
      GO TO 25
   24 COUNT(IA)=COUNT(IA)+1
   25 R=1
      IF(J .LT. (N-1)) GO TO 18
      WRITE(IN2,16) (COUNT(I),I=1,6)
   16 FORMAT(1H ,5X,'RUN',6I8)
      W=1.64
      Z=12.6
      B(1)=(1./6.)
      B(2)=(5./24.)
      B(3)=(11./120.)
      B(4)=(19./720.)
      B(5)=(29./5040.)
      B(6)=(1./840.)
      V=0.
      DO 44 I=1,IA
      DO 44 J=1,IA
   44 V=V+(COUNT(I)-N*B(I))*(COUNT(J)-N*B(J))*A(I,J)
      V=V/N
      IF(V .LE. Z .AND. V .GE. W) GO TO 55
      WRITE(IN2,65) V
   65 FORMAT(1H ,5X,'V =',E17.7,5X,'REJECT RUN')
      GO TO 77
   55 WRITE(IN2,66) V
   66 FORMAT(1H ,5X,'V =',E17.7,5X,'ACCEPT RUN')
   77 RETURN
      END
```

SUBROUTINE GAPT

SOURCE:

Knuth, Donald E., *The Art of Computer Programming*. Addison-Wesley Publishing Company, Inc. 1969.

PURPOSE:

To check if the length of NG gaps are distributed as expected in "true" random numbers.

CALLING SEQUENCE:

Random numbers between 0 and 1 are generated before GAPT is called.

CALL GAPT (N,JG,R,NG)

where:

N is the number of random numbers generated.

JG is the length of the longest gap being counted.

R is the array containing the random numbers.

NG is the number of gaps that are counted.

METHOD:

The first random number is compared with the following random numbers until it is found to be equal to one of the following random numbers. A gap is of length L, where L is the number of random numbers between those two equal random numbers. The next random number is used to compare with the following random numbers. The process is continued until NG gaps have been found.

$$EP(0) = P(NG), \qquad EP(I) = \sum_{I=1}^{JG-1} NG(1 - P)^{I},$$

$$EP(JG) = NG(P)(1 - P)^{JG}.$$

EP(I) is the expected number of gaps for a gap length of I.

The χ^2 statistic is then computed by:

$$\chi^2 = \sum_{I=0}^{JG} [GAP(I) - EP(I)]^2/EP(I).$$

χ^2 has approximately a chi-square distribution with JG degrees of freedom for "truly" random numbers.

COMMENTS:

The upper and lower limits for a 90% confidence interval have been put into the subroutine. χ^2 is then checked to see if it falls between these limits. The percent of confidence interval may be changed by changing the limits in the subroutine.

```
      SUBROUTINE GAPT(N,JG,R,NG)
C THIS IS THE GAP TEST.
C A GAP OF LENGTH K IS OBTAINED WHEN THERE ARE K DIGITS BETWEEN TWO
C DIGITS WHICH ARE IDENTICAL.
C N IS THE NUMBER OF RANDOM NUMBERS GENERATED FOR THIS TEST.
C JG IS THE LENGTH OF THE LONGEST GAP BEING RUN.
C ANY GAP LONGER THAN JG IS BEING COUNTED AS A GAP OF LENGTH JG.
C NG IS THE NUMBER OF GAPS.
      DIMENSION R(1),GAP(100),EP(100)
C  IN1 AND IN2 ARE LOGICAL DEVICE NUMBERS.  TEXAS A+I
C USES 1 TO READ AND 3 TO WRITE FOR THE IBM 360/44 COMPUTER.
      IN1=1
      IN2=3
      IS=0
      GAPO=0
      DO 5 I=1,JG
    5 GAP(I)=0
      K=R(1)*10+1
      L=0
      J=2
   19 L=0
      DO 6 I=J,N
      M=R(I)*10+1
      IF(M .EQ. K) GO TO 11
      L=L+1
      IF(L .EQ. JG) GO TO 11
      GO TO 6
   11 J=I+2
      K=R(I+1)*10+1
      IF (L .GE. JG) GO TO 17
      IF(L .NE. 0) GO TO 12
      GAPO=GAPO+1
      GO TO 18
   12 GAP(L)=GAP(L)+1
      GO TO 18
   17 GAP(JG)=GAP(JG)+1
   18 IS=IS+1
      IF(IS .EQ. NG) GO TO 22
      GO TO 19
    6 CONTINUE
      WRITE(IN2,13) IS
   13 FORMAT(1H ,'NOT ENOUGH RANDOM NUMBERS',10X,'IS =',I5)
      GO TO 88
   22 P=.1
      WRITE(IN2,211) GAPO
  211 FORMAT(1H ,'GAPO =',F4.1)
      WRITE(IN2,213) (GAP(I),I=1,10)
  213 FORMAT(1H ,'GAP =',10F8.1)
      EPO=P*NG
      K=JG-1
      DO 27 L=1,K
   27 EP(L)=NG*P*((1-P)**L)
      EP(JG)=NG*((1-P)**JG)
      CS=(GAPO-EPO)**2/EPO
      DO 77 I=1,JG
   77 CS=CS+(GAP(I)-EP(I))**2/EP(I)
      K=JG
      A=18.3
      B=3.94
```

```
      IF(CS .GE. B .AND. CS .LE. A) GO TO 26
      WRITE(IN2,51) K,A,B,CS
      GO TO 88
   26 WRITE(IN2,52) K,A,B,CS
   51 FORMAT(1H ,'90( CONF. INT. WITH K DEG. OF FREEDOM',5X,'K =',I3,5X,
     1'A =',F7.2,5X,'B =',F7.2,5X,'CS =',F10.2,5X,'REJECT GAP TEST')
   52 FORMAT(1H ,'90( CONF. INT. WITH K DEG. OF FREEDOM',5X,'K =',I3,5X,
     2'A =',F7.2,5X,'B =',F7.2,5X,'CS =',F10.2,5X,'ACCEPT GAP TEST')
   88 RETURN
      END
```

SUBROUTINE LPTEST

SOURCE:

Naylor, T. H., Balintfy, J. L., Burdick, D. S., and Chu Kong. Computer Simulation Techniques. New York: John Wiley and Son, Inc., 1966.

PURPOSE:

To check if there is a correlation between r_i and r_{i+k} random numbers.

CALLING SEQUENCE:

Random numbers between 0 and 1 are generated before LPTEST is called.

CALL LPTEST (N,R)

where:

N is the number of random numbers.

R is the array of random numbers.

METHOD:

The lagged product coefficient, C_k, is computed for each K. where K is the length of the lag.

$$C_k = \frac{1}{N - K} \sum_{i=1}^{N-K} r_i r_{i+k}.$$

If there is no correlation between r_i and r_{i+k}, the value of C_k will be approximately normally distributed with expected value of 0.25.

Lower and upper limits are computed for 90% confidence interval and each C_k is checked to see if it falls between these limits. Standard deviation is equal to $\sqrt{13N - 19K/12}(N - K)$.

COMMENTS:

The 90% confidence interval can be changed, by changing the value of z in the subroutine. Z and -z are the values for 90% confidence interval of normal distribution.

The value of K cannot be larger than N.

```
      SUBROUTINE LPTEST (N,R)
      DIMENSION R(1)
C  IN1 AND IN2 ARE LOGICAL DEVICE NUMBERS.  TEXAS A+I
C USES 1 TO READ AND 3 TO WRITE FOR THE IBM 360/44 COMPUTER.
      IN1=1
      IN2=3
      DO 99 K=1,15
      C=0.
      M=N-K
      DO 2 I=1,M
    2 C=C+R(I)*R(I+K)
      CK=C/M
      WRITE(IN2,100) K,CK
  100 FORMAT (1H ,'K =',I2,10X,'CK =',F10.5)
      Z = 1.64
      SD=SQRT(13.*N-19.*K)/(12.*M)
      B=0.25+Z*SD
      A=0.25-Z*SD
      WRITE(IN2,103) A,B
      IF (CK .GE. A .AND. CK .LE. B) GO TO 40
      WRITE(IN2,102)
      GO TO 99
   40 WRITE(IN2,101)
  101 FORMAT (1H ,'ACCEPT THE LAGGED PRODUCT TEST')
  102 FORMAT (1H ,'REJECT THE LAGGED PRODUCT TEST')
  103 FORMAT(1H ,'A =',F10.5,10X,'B =',F10.5)
   99 CONTINUE
   88 RETURN
      END
```

SUBROUTINE MATRIX

SOURCE:

Naylor, T. H., Balintfy, J. L., Burdick, D. S., and Chu Kong. Computer Simulation Techniques. New York: John Wiley and Son, Inc., 1966.

PURPOSE:

To determine if successive numbers are "truly" random.

CALLING SEQUENCE:

N random numbers between 0 and 1 are generated before MATRIX is called.

CALL MATRIX(N,R,L)

where:

N is the number of random numbers

R is the array containing the random numbers

L indicates that the size of the matrix is LxL.

METHOD:

The interval of 0 to 1 is divided into L subintervals. Successive random numbers are paired off and placed into an LxL matrix according to the random numbers of that pair. The χ^2 statistic is then computed as follows:

$$\chi^2 = \sum_{i=1}^{L} \sum_{j=1}^{L} (f_{ij} - E)^2/E$$

where f_{ij} is the number of pairs of random numbers in each element of the matrix and E is the expected number of pairs of random numbers in each element of the matrix. χ^2 has approximately a chi-square distribution with L*L - 1 degrees of freedom for "truly" random numbers.

COMMENT:

This subroutine calculates the upper and lower limits for the numbers to be accepted as "truly" random. Z and W are the chi-square values at 90% confidence interval with L*L - 1 degrees of freedom. The percent of confidence interval may be changed, by changing Z and W in the subroutine.

```
      SUBROUTINE MATRIX(N,R,L)
      DIMENSION MTRX(32,32), R(1)
C  IN1 AND IN2 ARE LOGICAL DEVICE NUMBERS.  TEXAS A+I
C USES 1 TO READ AND 3 TO WRITE FOR THE IBM 360/44 COMPUTER.
      IN1=1
      IN2=3
C THE MATRIX IS SET TO ZERO.
      DO 22 I=1,L
      DO 22 J=1,L
   22 MTRX(I,J)=0
C RANDOM NUMBERS ARE PAIRED OFF AND A COUNTER IS INCREMENTED
C ACCORDING TO WHERE THE RANDOM PAIR FIT IN THE MATRIX.
      DO 11 I=1,N,2
      KM=L*R(I)+1
      LM=L*R(I+1)+1
   11 MTRX(KM,LM)=MTRX(KM,LM)+1
C E IS THE EXPECTED NUMBER OF PAIR OF RANDOM NUMBERS TO BE
C FOUND IN EACH ELEMENT OF THE MATRIX.
      U=FLOAT(N)/2.
      E=U/(FLOAT(L)*FLOAT(L))
C THE CS STATISTIC HAS A CHI-SQUARE DISTRIBUTION WITH
C   L*L-1 DEGREES OF FREEDOM.
      CS=0.
      DO 12 I=1,L
      DO 12 J=1,L
   12 CS=CS+(MTRX(I,J)-E)**2/E
      Z=1.64
      W=-Z
      K=L*L-1
      AK=K
C  A IS THE LOWER AND B IS THE UPPER LIMIT FOR THIS TEST TO BE ACCEPTED.
      A=W*SQRT(2.*AK)+AK
      B=Z*SQRT(2.*AK)+AK
      WRITE(IN2,44) K,A,B,CS
      IF(CS .GE. A .AND. CS .LE. B) GO TO 28
      WRITE(IN2,38)
      GO TO 88
   28 WRITE(IN2,39)
   44 FORMAT(1H ,10X,'90( CONFIDENCE INTERVAL WITH K DEGREES OF FREEDOM'
     1,/,10X,'K =',I6,5X,'A =',F9.2,5X,'B =',F9.2,5X,'CS =',F10.2)
   38 FORMAT(1H ,20X,'REJECT MATRIX TEST')
   39 FORMAT(1H ,5X,'ACCEPT MATRIX TEST')
   88 RETURN
      END
```